Um estudo comparativo dos protocolos de dinheiro digital

Cesar Mindoff
cmindoff@santander.com.br
Especializando ESPM
Bel. Ciência Computação (ULBRA, 2001)

Vinicius Gadis Ribeiro
vribeiro@inf.ufrgs.br
Professor da UNIRITTER, UNILASALLE, FACENSA
Phd Student - UFRGS

This paper presents the results of a comparative study among digital money cryptography protocols. Some indicators have been defined to measure data and compare results. All data have been obtained in laboratory conditions.
Keywords: Cryptography, Cryptography protocols, Digital money, Comparative Study.

O presente trabalho relata os resultados de um estudo comparativo entre protocolos criptográficos de dinheiro digital – base para diversos esquemas atualmente empregados para substituir o dinheiro físico. Para tanto, foram definidos indicadores para efetuar as medições, e comparar os resultados. Os lados aqui levantados foram obtidos em condições de laboratório.
Palavras chave: Criptografia, Protocolos Criptográficos, Dinheiro Digital, Estudo Comparativo.

1. Introdução

Um protocolo criptográfico é um protocolo ou esquema onde, em algum passo desse último, se faz uso de criptografia [SCH96]. Através do uso desses protocolos, pode-se estabelecer uma chave de sessão, dividir ou compartilhar segredo, votar, empregar dinheiro digital, etc. Um dos esquemas de maior interesse para a sociedade é o de dinheiro digital: para tanto, devem ser respeitadas determinadas restrições, para não incorrer e fraudes – de qualquer das partes que trabalham no protocolo. Schneier apresenta quatro protocolos de dinheiro digital, aqui referidos como protocolos 1, 2, 3 e 4. As entidades que deles participam são tratados, genericamente, de Cliente, Banco ou Comerciante. Maiores detalhes dos protocolos, assim como as restrições com as quais devem ser trabalhados, podem ser constatados em [SCH96].

Apesar do interesse geral da sociedade, pouca produção tem sido realizada no sentido de se determinar os aspectos de desempenho entre os mesmos – observa-se, em Foo [FOO97], um estudo comparativo das **funcionalidades** de esquemas de dinheiro digital, sem a preocupação de **desempenho** dos protocolos que suportam tais esquemas.

O trabalho que se segue apresenta os resultados de um estudo comparativo que mediu não as funções exigidas de esquemas de dinheiro eletrônico, mas os seus requisitos de desempenho.

Segundo Pressman [PRE95], o teste de desempenho é idealizado para testar o desempenho de execução do esquema dentro do contexto de um sistema integrado. O teste de desempenho ocorre ao longo de todos os passos do processo de teste. Somente quando todos os elementos de sistema estão plenamente integrados, é que o desempenho real de um sistema pode ser verificado.

O teste de desempenho, freqüentemente, exige instrumentação de *hardware* e *software*. Ou seja, muitas vezes é necessário medir rigorosamente a utilização de recursos (por exemplo, ciclos de processador). A instrumentação externa pode monitorar intervalos de execução e registrar eventos (por exemplo, interrupções), quando estes ocorrerem, e amostrar estados de máquina em base regular. Ao instrumentalizar um programa, o realizador do teste pode descobrir situações que levam à degradação e possível falha do sistema.

A seguir, são apresentados os indicadores de desempenho – quais são, como foram medidos e sua relevância.

2. Indicadores selecionados

Dentre os diversos critérios indicadores que podem influenciar diretamente o desempenho, foram escolhidos três: *tempo de processamento*, *número de mensagens que devem ser trocadas* e o *tamanho total das mensagens que devem ser trocadas*.

O *Tempo de Processamento* nada mais é que o tempo gasto por cada parte envolvida no protocolo – Banco, Comerciante e Cliente – para conclusão deste protocolo. Muitas vezes, a detecção do evento e a geração da resposta são simples. O processamento das informações sobre o evento para determinar a resposta apropriada é que pode envolver algoritmos complexos e consumidores de tempo.

O segundo indicador testado foi o *número de mensagens que devem ser trocadas* entre o cliente ou consumidor com a entidade reguladora – por exemplo, o banco. A necessidade de muitas trocas de mensagens degrada o sistema, tornando-o, com freqüência, mais vulnerável a ataques – visto que um atacante pode agir nas portas (ataques à autenticação), ou na comunicação entre as portas (ataque ao protocolo).

A última variável medida foi o *tamanho total das mensagens trocadas* entre cada uma das entidades envolvidas no protocolo, quanto maior forem as mensagens, maior será a necessidade de velocidade de transmissão de dados para um desempenho que satisfaça o usuário.

Evidentemente, não se esgota aqui a gama de indicadores que podem ser empregados. Nos estudos que estão se seguindo, busca-se estabelecer **relações mais diretas** entre esses e novos indicadores.

3. Procedimentos de medição

O primeiro indicador medido, o *Tempo de Processamento*, mede o tempo de processamento em cada uma das partes envolvidas. Para a obtenção deste valor, foram colocados contadores de tempo entre cada espera de mensagem – pois o objetivo é calcular o tempo de processamento, excluindo o tempo de tráfego e de espera entre as partes envolvidas. No que se refere a sua relevância, o *Tempo de Processamento* define uma das principais necessidades que o dinheiro eletrônico hoje necessita: rapidez e agilidade, a fim de propiciar um menor transtorno as partes. Porém, este indicador pode apontar também a necessidade de poder computacional – ou recursos de hardware - , uma vez que se um primeiro protocolo leva menor tempo para executar todo o seu processamento que um segundo protocolo, em máquinas idênticas, o primeiro protocolo necessita de um menor poder computacional que o segundo.

O *número de mensagens trocadas* entre as partes envolvidas consiste de contadores simples. A cada troca de mensagens, cada uma das partes atualiza seus contadores, tanto de mensagens recebidas quanto de mensagens enviadas. Cada componente do protocolo possui contadores de mensagens recebidas e enviadas para as duas entidades que com ele interagem, por exemplo, o Cliente obtém primeiramente o total de mensagens que enviou e que recebeu do banco, obtendo assim o total de mensagens trocadas com o banco. Num segundo momento, ele obtém o número de mensagens que enviou e que recebeu do comerciante, obtendo também o número total de mensagens trocadas com o comerciante. Cada uma das partes executa o mesmo processo. Uma vez que a cada mensagem trocada, uma das partes fica parada (esperando receber aquela informação para então dar continuidade ao seu processamento), um maior número de mensagens trocadas representa um atraso considerável, pois é necessário que as partes estejam sincronizadas.

A medição do *tamanho total das mensagens trocadas* entre cada uma das entidades envolvidas no protocolo obedeceu o seguinte critério: em cada entidade envolvida, a cada

mensagem recebida, essa entidade calcula o tamanho da mensagem recebida e soma a um contador, que ao final do processamento contém o numero total de bytes recebidos. Assim, o Cliente obtém o tamanho total das mensagens que recebeu do Banco, e também o tamanho total das mensagens que recebeu do Comerciante. E assim as outras entidades procedem para a obtenção do tamanho total em bytes das mensagens trocadas em cada uma das partes. Quanto maior o número total de bytes trocados entre as entidades, maior será o tráfego gerado pelas entidades envolvidas no protocolo. Embora em constante mudança – grande disseminação de acesso à rede mundial de computadores por outras maneiras, como o rádio, *cable modem*, etc., a grande maioria dos usuários possui hoje pequeno poder de transferência de dados, sendo este um fator que influi diretamente no tempo necessário para que o protocolo seja finalizado. Assim sendo, um maior número de bytes trocados influi diretamente na necessidade de velocidade de transmissão de dados para um desempenho que satisfaça o usuário.

4. Considerações referentes à implementação

Para realizar o estudo comparativo, foram empregadas máquinas com mesmas configurações, e empregada uma rede local. Para a implementação dos protocolos, bem como dos elementos medidores, foi empregada a linguagem Java. Buscou-se respeitar as condições experimentais previstas em Ribeiro[RIB00].

A seguir, são apresentados os pontos críticos da implementação: a assinatura digital parcialmente cega e as classes mais relevantes.

O principal procedimento em qualquer implementação de esquemas de dinheiro eletrônico é o procedimento que realiza o protocolo criptográfico de assinatura digital parcialmente cega (*blind digital signature*). Após um aprofundado estudo de como funciona este protocolo e quais as principais tecnologias utilizadas hoje no mercado para sua implementação, optou-se pela utilização do algoritmo DES – dada a sua rapidez – para criptografia simétrica, e DSA para assinatura digital. A implementação da assinatura digital atendeu o seguinte roteiro: num primeiro momento o cliente criptografa todas as requisições de dinheiro eletrônico utilizando o algoritmo DES, para tanto, o cliente gera uma chave DES diferente para cada requisição, criptografa cada uma com uma chave diferente. A seguir o cliente envia todas as requisições criptografadas ao banco. O banco por sua vez, escolhe randomicamente um número entre o número total de requisições recebidas – essa será a requisição a ser assinada - e envia esse número ao cliente. O cliente, então, envia ao banco todas as chaves DES por ele geradas, exceto a correspondente àquele número randômico. O banco, então, descodifica todas as requisições de dinheiro, exceto aquela que ele não possui a chave, verifica se todas elas correspondem ao desejado, para então, se todas as requisições estiverem de acordo, assinar digitalmente com sua chave privada – utilizando o algoritmo DSA – a requisição restante que ainda está criptografada. No momento em que o Banco envia ao cliente esta requisição, o cliente possui uma requisição assinada pelo banco – o que pode ser verificado utilizando a chave pública deste – com conteúdo totalmente obscuro ao Banco, pois ele jamais teve acesso a essa requisição que ele assinou sem estar criptografada.

Uma série de classes foram construídas para implementação dos quatro esquemas de dinheiro eletrônico testados neste estudo. As principais classes desenvolvidas foram a classe Cliente, a classe AssinaturaCahvePub e a classe RequisicaoDin. Todas essas classes implementam a interface *serializable*, uma interface JAVA utilizada para serialização de objetos, única maneira de transferência de objetos por *sockets*.

A primeira delas, a classe Cliente, define o objeto cliente. Para tanto, possui uma série de variáveis que guardam os dados do cliente, são elas: nome do cliente, endereço do cliente,

conta corrente do cliente e o valor da requisição que este cliente deseja. Esta é a primeira classe utilizada no programa, ela é utilizada para criar o objeto cliente, com os dados relativos a este cliente, para que o cliente envie este objeto ao banco quando da requisição do dinheiro eletrônico. Com a posse deste objeto é que o banco possui a capacidade de identificar o cliente e conta de onde deve ser retirado o dinheiro e transformado em dinheiro eletrônico. Esta classe é utilizada somente na operação de saque.
A segunda classe que deve aqui ser descrita é a classe AssinaturaChavePub. Esta é a principal classe da implementação. Ela nada mais é do que a classe que cria o objeto requisição de dinheiro eletrônico assinada pelo banco, ou seja, o próprio dinheiro eletrônico. Ela possui os seguintes campos: valor da requisição e assinatura digital gerada pelo banco com a utilização do algoritmo de assinatura digital DSA. A medida em que os protocolos implementam mais serviços de segurança, o número de campos da classe cresce proporcionalmente. No caso do protocolo 4, essa classe possui ainda um campo que guarda o *string* único gerado randomicamente, e dez outros *strings*, metade deles gerados randomicamente, e a outra metade, são *strings* resultantes da operação XOR entre os primeiros cinco *strings* e o *string* de identificação do cliente, que será revelado caso este tente trapacear alguma das partes envolvidas no protocolo. Esta é a principal classe da implementação e é utilizada em todas as transações dos esquemas de dinheiro eletrônico.
Por fim, a terceira classe que deve aqui ser descrita, é a classe RequisicaoDin. Esta classe é utilizada para criar o objeto requisição de dinheiro eletrônico. Esta classe é utilizada somente na transação de saque. Este objeto é criado pelo cliente, e possui as seguintes variáveis: o valor da requisição, e a medida em que cresce a necessidade de segurança, ela passa a implementar novas variáveis. No caso do protocolo 4, possui ainda um campo que guarda o *string* único gerado randomicamente, e dez outros *strings*, metade deles gerados randomicamente, e a outra metade, são *strings* resultantes da operação XOR entre os primeiros cinco *strings* e o *string* de identificação do cliente, que será revelado caso este tente trapacear alguma das partes envolvidas no protocolo. O cliente cria n objetos deste tipo e envia todos ao banco, o banco escolhe um, e assina somente um, criando o outro objeto descrito anteriormente AssinaturaChavePub, que é o dinheiro eletrônico e o devolve ao cliente.

5. Resultados Obtidos

A seguir, são apresentados os resultados obtidos sobre as variáveis ou indicadores definidos previamente. As variáveis de interesse para o presente trabalho são o tempo de processamento, o número de mensagens a serem trocadas entre as partes envolvidas e o tamanho total das mensagens trocadas entre cada uma das partes envolvidas nos protocolos.
Tempo de processamento
O primeiro indicador a ser analisado é o tempo de processamento obtido para cada protocolo. A apresentação dos tempos obtidos segue um roteiro, mostrando inicialmente os tempos obtidos em cada uma das operações realizadas no protocolo – saque, pagamento e depósito – para cada um dos agentes envolvidos – Cliente, Banco e Comerciante. Por fim, são apresentados os tempos totais necessários para a execução completa de cada uma das operações em cada um dos protocolos, expondo também o tempo total gasto em cada protocolo.
A tabela a seguir apresenta os tempos necessários para o completo processamento da tarefa de cada uma das partes envolvidas no protocolo somente para a operação de saque.

Tabela 1 – Tempo de processamento - saque

Protocolo	$T_{Cliente}$ (ms)	T_{Banco} (ms)	Total (ms)
1	57852	63575	121427
2	59350	73670	133020
3	61958	74175	136133
4	63255	74390	137645

Analisando-se a tabela acima, observa-se que a medida em que existe uma maior necessidade de segurança no protocolo – com a revelação da parte que tenta trapacear e posteriormente podendo revelar a identidade deste indivíduo – sua complexidade aumenta, aumentando a necessidade de um tempo maior de processamento gasto por cada um dos agentes envolvidos neste protocolo.

A tabela abaixo apresenta os tempos necessários para o completo processamento da tarefa realizada por cada uma das entidades envolvidas no protocolo para a operação de pagamento.

Tabela 2 – Tempo de processamento - pagamento

Protocolo	$T_{Cliente}$ (ms)	T_{Banco} (ms)	Total (ms)
1	490	24140	24630
2	570	24760	25330
3	675	24875	25550
4	1155	31050	32205

Conforme pode ser visto na operação de saque (tabela 1), os tempos necessários para a conclusão da operação de depósito obedecem o mesmo comportamento – tempos crescentes conforme a necessidade de maior segurança no protocolo. Cabe ressaltar, porém, o tempo gasto pelo quarto protocolo em relação aos demais na operação de pagamento. O aumento substancial do tempo de processamento necessário deve-se, provavelmente, à necessidade de uma interação maior entre Cliente e Comerciante, havendo a necessidade do Comerciante escolher randomicamente quais partes dos *string*s de identificação revelar, tendo de informar este dado ao cliente para o mesmo assim proceder.

A próxima tabela a ser analisada expõe os tempos necessários para o completo processamento da tarefa de cada uma das partes envolvidas no protocolo para a execução da operação de depósito.

Tabela 3 – Tempo de processamento - depósito

Protocolo	$T_{comerciante}$ (ms)	T_{banco} (ms)	Total (ms)
1	570	1345	1915
2	553	1565	2118
3	527	1587	2114
4	1773	4270	6043

Essa tabela reflete o que já foi comentado. Conforme a crescente necessidade de segurança, crescem também os tempos de processamento necessários para a execução da operação. Novamente aqui, o tempo gasto pelo quarto protocolo em relação aos demais eleva-se bastante. O aumento substancial do tempo de processamento nesta operação pelo quarto protocolo deve-se, provavelmente, ao fato de haver a necessidade de um tratamento especial aos *string*s de identificação em relação aos outros protocolos. A necessidade do armazenamento de cinco *string*s de identificação para cada operação de depósito deve contribuir sensivelmente para tamanho aumento nos tempos obtidos.

Tabela 4 – Tempo total(ms) empregado por cada protocolo.

Protocolo	Saque	Pagamento	Depósito	Total
1	121427	24630	1915	147972
2	133020	25330	2118	160468
3	136133	25550	2114	163797
4	137645	32205	6043	175893

Essa tabela é bastante conclusiva. Conforme já foi definido anteriormente, os tempos crescem na medida em que o processamento necessário também cresce, e isso pode ser explicado analisando os passos realizados por cada protocolo. Outro dado bastante relevante, apresentado por essa tabela, é a diferença dos tempos de processamento necessários para a realização de cada operação. A operação de saque desponta com um número bastante significativo em relação às outras operações, podendo determinar aqui um gargalo, pois os tempos necessários para a operação de pagamento e depósito são substancialmente menores. Essa diferença deve-se ao número substancialmente maior de passos a serem realizados na operação de saque, apontando a maior necessidade de processamento nesta fase do protocolo.

Número de mensagens trocadas entre as entidades envolvidas

A seguir, são apresentados os resultados obtidos para o segundo indicador testado neste estudo: o número de mensagens total de cada protocolo é, na verdade, o número de mensagens trocadas entre cada uma das entidades envolvidas no protocolo – Cliente, Comerciante e Banco – em cada uma das operações realizadas – saque, pagamento e depósito.

Tabela 5 - Número total de mensagens trocadas em cada protocolo

Protocolo	Saque	Pagamento	Depósito	Total
1	43	5	7	55
2	43	5	7	55
3	43	5	7	55
4	43	15	7	65

A tabela acima mostra claramente que há um padrão de comportamento nos três primeiros protocolos, verificando-se alteração exclusivamente no quarto protocolo. Isso define um atraso maior neste último protocolo, no que se refere à necessidade de sincronismo entre as entidades envolvidas no protocolo, uma vez que a necessidade de uma maior troca de mensagens, representa uma espera maior por parte de quem espera a mensagem de retorno.

Tamanho das mensagens trocadas entre as entidades envolvidas

O último indicador testado neste estudo é o tamanho das mensagens trocadas entre as entidades envolvidas. A seguir, são apresentados os resultados obtidos para este último indicador, primeiramente entre cada uma das entidades envolvidas no protocolo – Cliente, Comerciante e Banco – em cada uma das operações realizadas – saque, pagamento e depósito.

Primeiramente, é apresentada a quantidade de bytes trocados entre Cliente e Banco para a realização da operação de saque.

Tabela 6 – Tamanho das mensagens trocadas - saque

Protocolo	Cliente -> Banco	Banco -> Cliente	Total
1	286	729	1015
2	606	830	1438
3	606	832	1440
4	3819	1152	4971

Observando-se a tabela acima, pode-se concluir que há um aumento significativo do número de bytes trocados no primeiro protocolo em relação ao segundo e ao terceiro, e um aumento muito maior pode ser observado no quarto protocolo. Isso se deve à necessidade do envio dos *string*s entre os protocolos. A diferença entre o primeiro protocolo, o segundo e o terceiro deve-se ao fato de que no segundo e terceiro protocolos existe a necessidade da troca do *string*

único randomicamente gerado, para fins de identificação de qual entidade do protocolo tentou trapacear no caso de alguma irregularidade. A diferença encontrada entre os protocolos 2 e 3 e o protocolo 4 deve-se, ainda, à necessidade da troca dos *string*s de identificação gerados, o que garante um tráfego bem maior de bytes, pois é necessária uma seqüência destes *string*s para a execução do protocolo de divisão secreta.

A seguir, pode ser avaliada a tabela que demonstra o número de bytes trocados entre Cliente e Comerciante para a realização da operação de pagamento.

Tabela 7 – Tamanho das mensagens trocadas (bytes) - pagamento

Protocolo	Cliente -> Comerciante	Comerciante -> Cliente	Total
1	63	172	235
2	94	172	266
3	126	172	298
4	444	182	626

De acordo com o que pode ser observado na operação de saque, a diferença entre o número de bytes enviados do Cliente ao Comerciante em cada um dos protocolos, deve-se à necessidade do envio dos *string*s entre os protocolos. O que deve ser observado aqui é a diferença que o quarto protocolo revela em relação aos demais, no que se refere aos bytes enviados do Comerciante ao Cliente. Essa diferença deve-se à necessidade do Comerciante enviar ao Cliente, no protocolo 4, quais *string*s de identificação o mesmo deve revelar, gerando um maior tráfego de bytes.

A próxima tabela expõe o número de bytes trocados entre Cliente e Comerciante para a realização da operação de depósito.

Tabela 8 – Tamanho das mensagens trocadas - depósito

Protocolo	Comerciante -> Banco	Banco -> Comerciante	Total
1	81	262	343
2	112	262	374
3	144	262	406
4	272	262	534

Essa tabela mostra o crescente número de bytes que deve ser enviado do Comerciante ao Banco, graças à necessidade do aumento de dados que devem ser enviados para uma apuração maior de quem está trapaceando. Pode-se observar, também, que o número de mensagens que o Banco envia ao Comerciante é invariante, uma vez que o Banco não necessita enviar dados relevantes ao consumidor - mas tão somente informá-lo se a operação foi ou não executada com sucesso.

Por fim, é apresentada a tabela que expõe o número total de bytes trocados entre os agentes envolvidos em cada um dos quatro protocolos analisados.

Tabela 9 – Número total de bytes trocados em cada protocolo

Protocolo	Saque	Pagamento	Depósito	Total
1	1015	235	343	1593
2	1438	266	374	2078
3	1440	298	406	2144
4	4971	626	534	6131

A tabela acima resume os resultados obtidos para este último indicador. Uma análise em cima dos resultados demonstra a crescente necessidade de tráfego de bytes na medida em que o nível de segurança exigido nos protocolos cresce. O aumento da segurança, nesses protocolos, é obtido através de uma troca maior de informações que podem revelar quem está tentando trapacear no caso de alguma irregularidade. Conclui-se, portanto, que o aumento substancial do número de bytes trocados no quarto protocolo se deve à necessidade de um número maior de informações trocadas, a fim de atingir a segurança a que este protocolo se propõe.

6. Considerações Finais e estudos futuros

Com o decorrer da pesquisa realizada para a realização do presente estudo, a importância da tecnologia dinheiro eletrônico ficou cada vez mais evidente. Embora ainda não disseminadas entre o público, algumas formas de pagamento eletrônico, como o cartão de crédito, implementações de dinheiro eletrônico já são uma realidade e vêm sendo utilizadas com sucesso por algumas empresas, por exemplo o *e-cash*, da empresa holandesa *Digicash*, sendo considerada hoje a forma de pagamento na rede mais parecida com o dinheiro real.

Assim, estudar o desempenho e o comportamento dos protocolos que os suportam tem elevada relevância.

A seguir, é apresentada uma tabela contendo totalizadores de todos os dados obtidos em relação a cada um dos quatro protocolos.

Tabela 10 – Desempenho geral

Protocolo	Tempo de Processamento (ms)	Número Total de Mensagens Trocadas	Tamanho Total(bytes) das Mensagens Trocadas
1	147972	55	1593
2	160468	55	2078
3	163797	55	2144
4	175893	65	6131

A tabela é bastante conclusiva. Levando-se em consideração o conhecimento do esquema proposto por cada protocolo em relação aos resultados obtidos em cima dos indicadores definidos, algumas considerações podem ser facilmente levantadas. Primeiramente, observa-se uma diferença muito pequena nos resultados obtidos entre os protocolos um, dois e três. Levando-se em consideração o grau de segurança que cada um deles proporciona, fica fácil optar pelo protocolo número três dentre essas opções. A segunda análise que deve ser realizada, é quanto à diferença dos dados obtidos para o protocolo quatro em relação aos demais. Devido ao maior grau de complexidade desse protocolo, os números obtidos para o mesmo ficaram bastante acima dos demais. Para uma definição entre qual protocolo seria mais interessante utilizar dentre o protocolo três e o protocolo quatro, é fundamental a observação das necessidades das partes envolvidas, levando-se em consideração o nível de segurança que se faz necessário em relação ao desempenho desejado, bem como a identificação de fraudadores.

Em suma, caso a identificação de fraudadores não seja o objetivo principal do esquema a ser desenvolvido, vale a pena, em termos de desempenho, empregar o protocolo 3, ao invés do protocolo 4.

Bibliografia

[FOO97] FOO, E.; BOYD,C.; CAELLI, W.; DAWSON, E. A Taxonomy of Electronic Cash Schemes In **Proceedings** of IFIP/SEC `97 13th International Information Security Conference, pages 337-348. Chapman and Hall, 1997.

[PRE95] PRESSMAN, R. S. **Engenharia de Software**, 3. Ed. – São Paulo: Makron Books, 1995. 1056 p.

[RIB00] RIBEIRO, V. G. **Um estudo sobre métodos de pesquisa utilizados em segurança computacional - Criptografia**, Porto Alegre: PPGC da UFRGS, 2000. 70 P. – (TI 916). Disponível na internet, em <http://www.sinpro-rs.org.br/vinicius.gadis.ribeiro>.

[SCH96] SCHNEIER, B**. Applied Cryptography: Protocols, Algorithms, and Source Code in C**. 2. Ed. – New York: John Wiley & Sons, 1996. 758 p.il.